Detecção Automatizada de Desinformadores em Grupos Públicos do WhatsApp Utilizando Redes Neurais para Grafos

Saulo Cavalcantel[1], Mikael Sales[1], Dagoberto Silva[1], Fernanda Ferreira[1], Wellington Franco[2], Rennan Dantas[2], José M Monteiro[2]

[1] Universidade Federal do Ceará (UFC) – Campus Crateús
Departamento de Computação – Universidade Federal do Ceará (UFC)

{saulocavalcante,mikaelmfsales,dagobertosilva,fernandaferreira}@alu.ufc.br

wellington@crateus.ufc.br, rennan@ufc.br, monteiro@dc.ufc.br

Abstract. *In recent years, the dissemination of misinformation through digital platforms has compromised public health, social stability, and even democratic processes. In the Brazilian context, WhatsApp is widely recognized as one of the main sources of misinformation. This study proposes an approach to automatically detect misinformation spreaders on WhatsApp by formulating the problem as a user classification task based on their interactions within the network. To this end, users and their connections are modeled as a graph. Subsequently, a neural network architecture leverages the previously constructed graph to capture structural patterns associated with the spread of misinformation. In this work, the GCN, GCNII, GraphSAGE, GAT, and APPNP models are evaluated stratified cross-validation using accuracy, precision, recall, F1-score, and AUC-ROC as performance metrics. The results show that GNNs can capture topological patterns relevant to identifying misinformation spreaders, indicating their potential to combat misinformation in digital environments.*

Resumo. *Nos últimos anos, a disseminação de desinformação por meio das plataformas digitais tem comprometido a saúde pública, a estabilidade social e até mesmo os processos democráticos. No contexto brasileiro, o aplicativo de mensagens WhatsApp é reconhecidamente uma das principais fontes de disseminação de desinformação. Este trabalho propõe uma abordagem para a detecção automatizada de desinformadores no WhatsApp, tratando o problema como uma tarefa de classificação de usuários com base em suas interações na rede. Para isso, os usuários e suas conexões são modelados como um grafo. Em seguida, uma arquitetura de rede neural explora o grafo previamente construído para capturar padrões estruturais associados à propagação de desinformação. Neste trabalho, os modelos GCN, GCNII, GraphSAGE, GAT e APPNP são avaliados por meio da validação cruzada estratificada, utilizando as métricas de acurácia, precisão, revocação, F1-score e AUC-ROC. Os resultados mostram que as GNNs capturam padrões topológicos relevantes para a identificação de desinformadores, o que indica seu potencial para o combate à desinformação em ambientes digitais.*

1. Introdução

As redes sociais digitais têm transformado profundamente os processos informacionais ao possibilitarem o acesso rápido e de baixo custo a conteúdos em larga escala

[Raudeliuniene et al. 2021]. Ao mesmo tempo, sua arquitetura algorítmica, associada ao caráter descentralizado e pouco regulado dessas plataformas, favorece a disseminação acelerada de desinformação e amplia seus impactos sociais [Metzler and Garcia 2024, Ferreira et al. 2021b]. Nesse contexto, destacam-se os desinformadores, entendidos como usuários que propagam conteúdos falsos ou enganosos de forma intencional e recorrente [Su et al. 2020].

A identificação desses agentes tem sido amplamente investigada na literatura por meio de modelos preditivos, análises comportamentais e abordagens baseadas em estruturas de rede, com especial atenção a plataformas mais abertas, como o *X*, nas quais há maior disponibilidade de dados históricos [DeVerna et al. 2024, Khan et al. 2023, Rath et al. 2022]. No entanto, no contexto brasileiro, o *WhatsApp* desempenha um papel central na circulação de informações e na comunicação cotidiana da população [Pereira et al. 2022]. Essa ampla adoção também favorece a propagação de conteúdos desinformativos, especialmente em contextos sensíveis, como crises sanitárias e períodos eleitorais [Fiocruz 2020, Ferreira et al. 2021a].

Apesar de esforços recentes voltados à detecção de conteúdos desinformativos e de seus disseminadores no *WhatsApp* [Cabral et al. 2021], esse problema permanece em aberto, principalmente quanto à utilização de dados associados à estrutura relacional entre participantes de grupos públicos [Nobre et al. 2022]. Em especial, o estudo de [Cabral et al. 2023], que constitui a referência mais próxima desta pesquisa, apresenta resultados relevantes quanto a atributos textuais, comportamentais e sociais, mas não explora redes neurais para grafos, com a finalidade de modelar diretamente a topologia de interação entre usuários. Assim, investigar em que medida a modelagem relacional da rede pode contribuir para a detecção automatizada de desinformadores no WhatsApp continua sendo uma questão fundamental.

Diante desse cenário, este artigo propõe uma abordagem para a identificação automatizada de desinformadores em grupos públicos do *WhatsApp* com base em Redes Neurais para Grafos (*Graph Neural Networks* - GNN), modelando usuários e interações como um grafo dirigido e ponderado construído a partir do conjunto de dados *FakeWhatsApp.Br*. Nessa representação, as arestas são reconstruídas com base na coocorrência em grupos e ponderadas de acordo com a atividade dos usuários e o grau de viralização das interações, permitindo comparar cinco arquiteturas distintas: GCN, GCNII, GraphSAGE, GAT e APPNP.

O objetivo deste trabalho é identificar e classificar agentes disseminadores de desinformação em um cenário fortemente desbalanceado, via protocolo experimental com validação cruzada estratificada, particionamento em treino, validação e teste, e avaliação por meio de métricas como acurácia, precisão, *recall*, *F1-score* e AUC-ROC. Os resultados indicam que a modelagem relacional da rede é informativa para a tarefa investigada, com destaque para GraphSAGE e GCNII no equilíbrio entre precisão e sensibilidade, além de fornecer uma base metodológica para o estudo da propagação de desinformação em plataformas fechadas e sem conexões sociais explicitamente observáveis.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica, a Seção 3 discute os trabalhos relacionados, a Seção 4 descreve a metodologia proposta, a Seção 5 apresenta e analisa os resultados experimentais, e, por

fim, a Seção 6 reúne as conclusões e os direcionamentos para trabalhos futuros.

2. Fundamentação Teórica

Conforme [Su et al. 2020], desinformação refere-se à disseminação, geralmente intencional, de informações deturpadas, englobando conteúdos falsos, enganosos, imprecisos ou distorcidos, elaborados com o propósito de atingir objetivos específicos. Com a popularização das redes sociais, esse fenômeno passou a ocorrer em maior escala, uma vez que essas plataformas ampliaram significativamente o alcance e a velocidade de propagação das informações. Nesse sentido, [Vosoughi et al. 2018] destacam que as redes sociais não apenas facilitam, mas também intensificam a disseminação da desinformação, dificultando a distinção entre conteúdos verdadeiros e falsos.

Esse cenário é agravado pelo fato de que, nessas plataformas, o conteúdo pode ser criado e compartilhado por qualquer usuário sem a necessidade de verificação prévia, contribuindo para a rápida propagação de informações duvidosas. Como consequência, torna-se fundamental compreender o papel dos usuários nesse processo. Nesse contexto, adota-se a definição de [Chu et al. 2010], segundo a qual um desinformador é caracterizado por apresentar comportamento anômalo em relação aos demais usuários, envolvendo não apenas a disseminação de conteúdos enganosos, mas também o uso estratégico de mecanismos como viralidade e personalização para ampliar seu alcance e influência.

Diante desse problema, a identificação de agentes de desinformação tem sido amplamente investigada na literatura. Segundo [Shu et al. 2017], essa tarefa baseia-se na aplicação de técnicas de análise de dados, aprendizado de máquina e verificação fática, com o objetivo de detectar padrões comportamentais atípicos e identificar os propagadores de conteúdos enganosos.

Um grafo G é definido como um par $G = (V, E)$, em que V representa o conjunto de vértices e E o conjunto de arestas que conectam esses vértices [West et al. 2001] e [Bondy et al. 1976]. No grafo direcionado (digrafo), as arestas possuem orientação, indicando uma relação de origem e destino entre vértices. Já no grafo não direcionado, as arestas não possuem direção, representando relações bidirecionais entre os vértices.

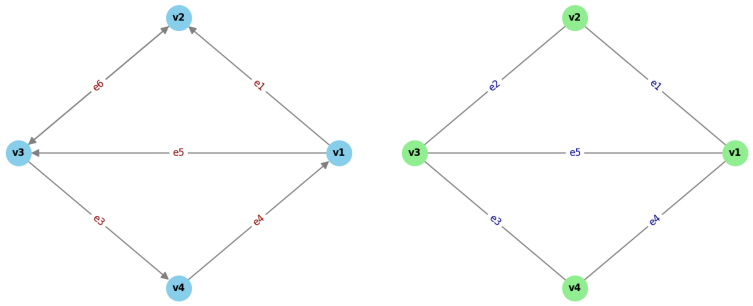

Figura 1. Exemplo de Grafo Direcionado (à esquerda) e Grafo Não Direcionado (à direita).

De acordo com [Easley et al. 2010], modelagens baseadas em grafos permitem analisar propriedades de redes sociais, como centralidade, formação de comunidades e propagação de informações, contribuindo para a compreensão das dinâmicas sociais.

Em redes como o *WhatsApp*, os usuários podem ser representados como vértices e suas interações como arestas. Em grafos não direcionados, uma aresta indica a troca mútua de mensagens, enquanto, em grafos direcionados, a direção da aresta indica o fluxo da comunicação entre os usuários.

As Redes Neurais em Grafos (*Graph Neural Networks* - GNNs), constituem uma classe de modelos de aprendizado profundo desenvolvida para processar dados estruturados em forma de grafo. Diferentemente das redes neurais tradicionais, que operam sobre dados euclidianos, como imagens e sequências, as GNNs são capazes de explorar explicitamente a estrutura relacional existente entre entidades conectadas, incorporando informações tanto dos atributos dos nós quanto de sua vizinhança no grafo. Em geral, esses modelos funcionam por meio de mecanismos iterativos de propagação e agregação de informação, nos quais cada nó atualiza sua representação vetorial com base em suas próprias características e nas de seus vizinhos. Dessa forma, as GNNs permitem capturar padrões topológicos e dependências estruturais complexas, sendo amplamente aplicadas em tarefas como classificação de nós, predição de links, detecção de comunidades, recomendação de itens e análise de redes sociais [Easley et al. 2010].

3. Trabalhos Relacionados

O estudo apresentado por [DeVerna et al. 2024] investiga a identificação e caracterização de *superspreaders* de conteúdo de baixa credibilidade no *X*. A abordagem baseia-se em métricas quantitativas de influência e volume de compartilhamento, complementadas por análise estatística das características desses usuários. Os resultados mostram que um grupo reduzido de contas, incluindo influenciadores políticos e veículos de mídia associados à desinformação, concentra grande parte da viralização desse conteúdo. O trabalho também observa associação entre linguagem mais tóxica e maior engajamento. Como limitação, a identificação dos *superspreaders* depende de métricas manuais e não explora modelos de aprendizado capazes de capturar padrões relacionais mais complexos.

Em [Khan et al. 2023], o problema é tratado como uma tarefa de forense comportamental em redes sociais, com o objetivo de distinguir usuários que disseminam desinformação deliberadamente, usuários que a propagam sem intenção e agentes que promovem refutação. Para isso, os autores combinam técnicas de aprendizado de máquina, como regressão logística e *Naive Bayes*, com *embeddings* de grafos, explorando estruturas relacionais, padrões temporais e interações entre usuários. Os resultados indicam precisão de 77,45% e *recall* de 75,80% na detecção de atores maliciosos, além de sugerir que parte relevante da disseminação ocorre por meio de usuários ingênuos. Contudo, o método apresenta menor eficácia em perfis pouco ativos e foi desenvolvido no contexto do *X*, o que limita sua transferência direta para plataformas com dinâmica distinta, como o *WhatsApp*.

O trabalho de [Rath et al. 2022] propõe uma estratégia preditiva para identificar *spreaders* de *fake news* em redes sociais com base em relações de confiança e topologia da rede. O *framework* utiliza GNN em regime indutivo e incorpora métricas como *trustingness* e *trustworthiness*, além de mecanismos auxiliares para estimar credibilidade das arestas e vulnerabilidade estrutural das comunidades. Os resultados mostram acurácia de 93,3% na identificação de disseminadores, com ganho adicional de até 4,6% quando há filtragem de *bots*, evidenciando a relevância da estrutura relacional para a tarefa. Ainda

assim, a abordagem depende fortemente de histórico de atividade e de informações observáveis da rede, o que reduz sua aplicabilidade em ambientes mais fechados e criptografados.

O estudo de [Cabral et al. 2023] é o mais próximo desta pesquisa por tratar diretamente do *WhatsApp*. Os autores introduzem o *dataset FakeWhatsApp.Br*, construído a partir de 59 grupos públicos monitorados entre julho e novembro de 2018, totalizando 282.601 mensagens e 5.364 usuários. A metodologia combina atributos textuais, comportamentais e sociais, com uso de TF-IDF e classificadores supervisionados, para detectar desinformação e identificar desinformadores em português brasileiro. Os resultados mostram desempenho elevado, com destaque para a regressão logística. Apesar disso, o estudo não modela explicitamente a rede de usuários por meio de redes neurais em grafos, o que deixa em aberto a exploração direta da topologia de interação como fonte principal de informação para a classificação.

Em síntese, os trabalhos analisados demonstram que atributos comportamentais, estruturais e topológicos são úteis para identificar agentes associados à disseminação de desinformação. No entanto, observa-se uma lacuna específica no contexto do *WhatsApp*: a aplicação de redes neurais em grafos para classificar usuários a partir da estrutura de interação reconstruída entre participantes de grupos públicos. Este trabalho avança nesse ponto ao representar a rede como grafo e avaliar arquiteturas de GNN para a identificação de desinformadores nessa plataforma.

4. Metodologia

Esta seção descreve a metodologia para identificar desinformadores em grupos públicos do *WhatsApp* usando ML e GNN. Nesse contexto, o problema é tratado como a identificação de usuários cujos padrões de interação e de difusão os diferenciam dos demais participantes da rede. A proposta de solução parte da hipótese de que disseminadores de desinformação apresentam um comportamento relacional distinto, passível de ser capturado por meio de uma representação em grafo. O método é organizado em quatro etapas: obtenção dos dados brutos, pré-processamento dos atributos, construção do grafo de interações e treinamento de diferentes arquiteturas de redes neurais para grafos. Por fim, os modelos são avaliados com base em métricas de classificação, com ênfase em acurácia e F1-score, para identificar qual abordagem apresenta melhor desempenho na identificação desses agentes. A Figura 1 apresenta o fluxo geral da metodologia utilizada.

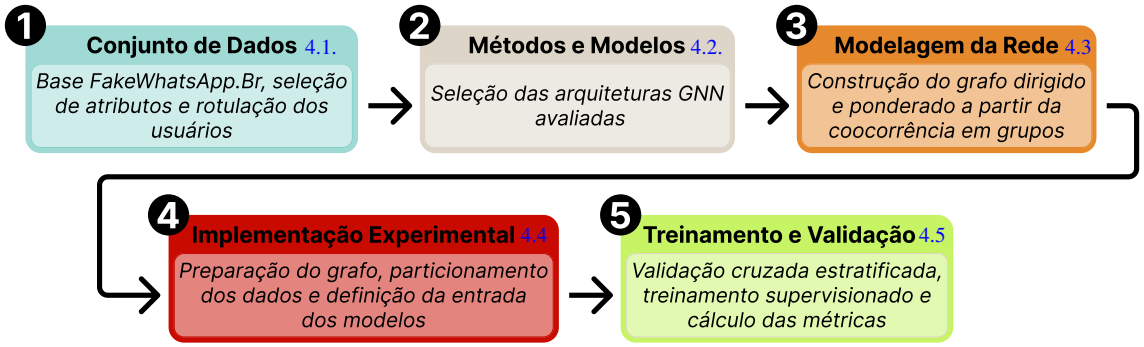

Figura 2. Etapas da Metodologia Utilizada.

4.1. Conjunto de Dados

O conjunto de dados utilizado neste trabalho deriva da dissertação de [Cabral et al. 2023] e do repositório *FakeWhatsApp.Br*[1]. O arquivo base empregado foi o ***users***, que reúne informações de 5.364 usuários de grupos públicos do *WhatsApp*. A partir desse arquivo, foi construído o conjunto ***users_selected_features***, utilizado como base para a modelagem dos nós do grafo. Para manter a aderência ao cenário de aplicação real, foram preservados apenas atributos potencialmente observáveis na própria plataforma, excluindo variáveis que dependem de anotações externas ou de conhecimento não nativo do *WhatsApp*. Além dos atributos individuais dos usuários, foram mantidas as informações de pertencimento a grupos, pois elas são necessárias para a reconstrução das relações entre participantes na etapa de modelagem da rede.

A rotulação dos usuários seguiu o critério de [Cabral et al. 2023], baseado na força viral (*viral_strength*) das mensagens. Neste trabalho, desinformador é definido operacionalmente como o usuário com *viral_strength* maior ou igual a 5.675, limiar herdado do trabalho de referência para manter a comparabilidade experimental. Com esse critério, 132 dos 5.364 usuários foram rotulados como desinformadores, correspondendo a aproximadamente 2,46% da base, enquanto 97,54% foram classificados como não desinformadores. Esse desbalanceamento influencia diretamente as etapas de particionamento, treinamento e avaliação dos modelos. A Figura 3 apresenta essa distribuição.

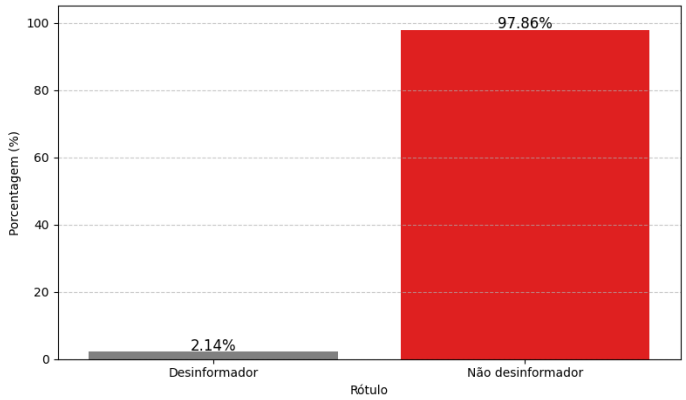

Figura 3. Proporção de Desinformadores vs Não Desinformadores.

4.2. Modelos Investigados

Foram avaliadas cinco arquiteturas de GNN para a tarefa de identificação de desinformadores no *WhatsApp*: GCN, GCNII, GraphSAGE, GAT e APPNP. A escolha desses modelos se baseia em sua ampla adoção e eficácia em tarefas de aprendizado sobre grafos, além de cobrirem estratégias complementares de propagação de informação na rede [Wu et al. 2020]. No contexto deste trabalho, essas arquiteturas foram empregadas para classificação de nós em um grafo dirigido e ponderado, no qual os usuários são representados por atributos comportamentais e relacionais, e as conexões entre eles representam interações inferidas a partir do compartilhamento de grupos.

Em conjunto, os modelos selecionados permitem comparar diferentes mecanismos de aprendizado sobre a estrutura do grafo, incluindo convolução espectral, propagação

[1] `https://github.com/cabrau/FakeWhatsApp.Br`

profunda com mitigação de *oversmoothing*, agregação por amostragem de vizinhança, atenção sobre conexões relevantes e propagação baseada em *PageRank*. Essa diversidade é pertinente para analisar como diferentes formas de agregação e propagação afetam a identificação de desinformadores em uma rede com forte desbalanceamento entre classes e relações reconstruídas indiretamente. A Tabela 1 resume os modelos avaliados e seus principais diferenciais.

Tabela 1. Modelos de GNN Selecionados.

Modelo	Descrição	Destaques
GCN	*Baseline* simples e eficaz para grafos médios.	Fácil implementação e bom desempenho.
GCNII	Versão aprimorada do GCN para redes profundas.	Evita *oversmoothing* e mantém acurácia.
GraphSAGE	Agrega vizinhos com amostragem eficiente.	Alta escalabilidade.
GAT	Aplica atenção às conexões relevantes.	Melhor desempenho em grafos heterogêneos.
APPNP	Propagação com PageRank personalizado.	Reduz *oversmoothing* e *overfitting*, permitindo maior profundidade da rede.

4.3. Modelagem da Rede

A rede de usuários foi modelada como um grafo dirigido e ponderado, em que cada nó representa um usuário e cada aresta representa uma interação inferida a partir da participação em grupos públicos do *WhatsApp*, conforme [Cabral et al. 2023]. Diferentemente de plataformas com conexões explícitas, como *X* e *Facebook*, no *WhatsApp* a estrutura relacional precisa ser reconstruída a partir da coocorrência dos usuários nos grupos. Neste trabalho, adotou-se o grafo de mensagens gerais, por representar de forma mais próxima o cenário real de uso da plataforma e por não depender de categorias específicas de conteúdo. A Tabela 2 resume os tipos de grafo disponíveis no conjunto de dados.

Tabela 2. Características e Descrição dos Grafos de Mensagens.

Grafo	Descrição	Nós	Arestas
Mensagens gerais	Cada nó é um usuário. Há aresta $i \to j$ se i enviou mensagem a um grupo do qual j participa, com peso igual ao total de mensagens enviadas por i.	5.364	1.125.326
Mensagens virais	Aresta $i \to j$ existe se i enviou mensagem viral a um grupo do qual j participa, ponderada pela quantidade de mensagens virais.	5.364	551.069
Mensagens com desinformação	Aresta $i \to j$ existe se i enviou mensagem com desinformação a um grupo do qual j participa, ponderada pela quantidade dessas mensagens.	5.364	433.204

Os atributos dos nós foram definidos a partir do conjunto *users_selected_features*, preservando apenas informações potencialmente observáveis no próprio *WhatsApp*, sem recorrer a fontes externas. Essa decisão busca manter a aderência ao cenário de aplicação e evitar a dependência de variáveis indisponíveis no uso real. Além dos atributos individuais, manteve-se a informação de pertencimento a grupos, necessária para reconstruir a estrutura relacional entre os usuários. A Tabela 3 apresenta os atributos utilizados na representação dos nós.

Tabela 3. Descrição dos Atributos Utilizados.

Feature	Descrição
`groups`	Grupos dos quais o usuário participa.
`number_of_messages`	Quantidade total de mensagens enviadas.
`texts`	Total de mensagens de texto enviadas.
`text_ratio`	Proporção de mensagens de texto em relação ao total de mensagens.
`midia`	Objetos multimídia compartilhados.
`midia_ratio`	Proporção de objetos multimídia em relação ao total de mensagens.
`virals`	Interações classificadas como virais.
`repeated_messages`	Mensagens repetidas enviadas.
`strenght`	Intensidade das interações do usuário.
`viral_strenght`	Grau de viralização das interações.

A construção das arestas considerou pares ordenados de usuários que compartilham ao menos um grupo. O peso de cada aresta foi definido em função da atividade do emissor e de sua força viral, de modo a representar a intensidade potencial de influência entre participantes que coexistem em grupos comuns. Assim, o peso foi calculado por:

$$\text{weight} = \text{num\_messages} \times \frac{\text{viral\_strength}}{1000} \tag{1}$$

Quando $\text{viral\_strength} = 0$, adotou-se $\varepsilon = 0{,}001$ para evitar pesos nulos. Caso um mesmo par de usuários compartilhasse mais de um grupo, os pesos eram acumulados, resultando em um grafo dirigido e ponderado que incorpora, simultaneamente, atividade, viralidade e coocorrência entre grupos. A Figura 4 ilustra uma fração do grafo de mensagens gerais, evidenciando sua densidade e a distribuição dos usuários na rede.

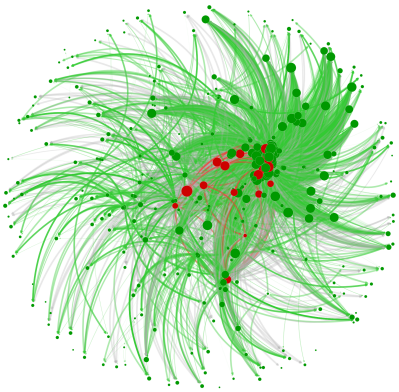

Figura 4. Demonstração de uma fração de 8% do grafo geral

4.4. Implementação dos Modelos de GNN

Os modelos foram implementados em `Python`, com o uso do `NetworkX`[2] para a construção do grafo, `scikit-learn`[3] para o particionamento e métricas, e `PyTorch`[4]/`PyTorch Geometric`[5] para o treinamento e a inferência das GNNs. O pipeline experimental foi organizado em quatro etapas: construção do grafo, particionamento em treino, validação e teste; definição das arquiteturas, e avaliação por meio de métricas de classificação. Essa implementação permite representar explicitamente a estrutura relacional dos usuários e aplicar aprendizado supervisionado sobre os nós do grafo.

O grafo foi construído a partir de um arquivo *CSV* em que cada linha representa um usuário e os grupos dos quais ele participa. Para cada grupo, foram identificados seus participantes e criadas arestas dirigidas entre cada par ordenado de usuários que compartilham esse grupo. Assim, a relação modelada não representa uma conexão social explícita, mas uma possibilidade de interação inferida pela coocorrência em grupos públicos. Os atributos dos nós foram extraídos do conjunto *users_selected_features*, mantendo apenas variáveis compatíveis com o contexto do *WhatsApp*.

O peso de cada aresta seguiu a formulação definida na Equação 1. Na implementação, quando a força viral era nula, adotou-se $\varepsilon = 0{,}001$ para evitar pesos nulos. Além disso, quando um mesmo par ordenado de usuários compartilhava mais de um grupo, os pesos eram acumulados. Com isso, preservou-se um grafo dirigido e ponderado, em que os nós representam usuários, as arestas representam relações inferidas a partir da exposição em grupos e os pesos representam a intensidade potencial de influência entre esses participantes.

Após a construção, o grafo foi convertido para a estrutura de dados utilizada pelo `PyTorch Geometric`, incluindo a matriz de atributos dos nós, o índice de arestas e o vetor de pesos associados. Essa representação foi empregada como entrada padrão para todas as arquiteturas avaliadas, garantindo a comparabilidade entre os modelos.

O particionamento foi realizado com estratificação por classe e distribuição de nós semelhantes entre os *splits*, de modo a reduzir concentração de perfis muito parecidos em um único subconjunto. Para isso, aplicou-se *KMeans* sobre os atributos dos nós, gerando agrupamentos utilizados em conjunto com os rótulos binários no processo de divisão. Essa estratégia busca aumentar a diversidade topológica entre treino, validação e teste e reduzir viés decorrente de redundância local.

Em cada agrupamento, os nós foram separados por classe, embaralhados e divididos segundo as proporções definidas para cada conjunto. Os índices resultantes foram convertidos em máscaras booleanas (`train_mask`, `val_mask` e `test_mask`) e utilizados no treinamento e na avaliação. Esse procedimento mantém a consistência entre a estrutura global do grafo e os subconjuntos usados em cada etapa do experimento.

Os modelos de GNN seguem a formulação usual de aprendizado em grafos, na qual cada nó atualiza sua representação a partir de seus próprios atributos e das

[2]NetworkX
[3]Scikit-learn
[4]PyTorch
[5]PyTorch Geometric

informações agregadas de sua vizinhança. Ao longo das camadas, esse processo produz *embeddings* que incorporam informações locais e estruturais do grafo, permitindo classificar usuários com base em seus padrões de atividade e posição relacional na rede.

As arquiteturas avaliadas neste trabalho compartilham essa estrutura geral, mas diferem no mecanismo de convolução, na forma de agregação da vizinhança e na profundidade de propagação. O *GCN* foi utilizado como *baseline* de convolução em grafos; o *GCNII* foi incluído por mitigar *oversmoothing* em redes mais profundas; o *GraphSAGE* explora agregação com amostragem de vizinhos; o *GAT* atribui pesos distintos às conexões por meio de atenção; e o *APPNP* combina predição neural com propagação inspirada em *PageRank*. Em conjunto, essas arquiteturas permitem comparar diferentes estratégias de propagação e agregação na tarefa de identificação de desinformadores.

4.5. Treinamento e Validação

A avaliação dos modelos foi conduzida por validação cruzada estratificada do tipo K-Fold, com $k = 5$, de modo a preservar, em cada partição, a proporção entre desinformadores e não desinformadores. Em cada iteração, um *fold* foi reservado para teste e os demais foram utilizados para treino e validação. Sobre esse subconjunto, aplicou-se uma divisão estratificada adicional de 80% para treino e 20% para validação. Para cada partição, foram geradas máscaras booleanas mutuamente exclusivas (`train_mask`, `val_mask` e `test_mask`), que definem os subconjuntos de nós utilizados em cada etapa do experimento. Esse procedimento permite avaliar os modelos em múltiplas divisões da base, reduzindo a dependência de uma única partição e tornando a estimativa de desempenho mais robusta.

A cada *fold*, uma nova instância do modelo foi inicializada e treinada com o otimizador `Adam`. Para lidar com o forte desbalanceamento entre as classes, utilizou-se a função de perda `NLLLoss` com pesos inversamente proporcionais à frequência das classes no conjunto de treino. O processo de treinamento empregou *early stopping* com base na perda de validação, preservando os parâmetros do melhor estado do modelo ao longo das épocas. Ao final de cada iteração, o modelo selecionado foi avaliado no conjunto de teste por meio das métricas de acurácia, precisão, *recall* e F1-score. As métricas finais de cada arquitetura foram obtidas pela média dos resultados nos cinco *folds*, permitindo comparar os modelos com base em seu desempenho médio no problema de classificação dos usuários.

5. Resultados

A Tabela 4 e a Figura 5 sintetizam o desempenho dos modelos avaliados. Na validação cruzada estratificada com $k = 5$, os melhores resultados médios foram obtidos por *GraphSAGE* e *GCNII*. O *GraphSAGE* apresentou *Accuracy* de 0,9965, *Precision* de 0,9682, *Recall* de 0,8855, *F1-Score* de 0,9222 e *AUC-ROC* de 0,9997. O *GCNII* obteve desempenho próximo, com *F1-Score* de 0,9089 e *AUC-ROC* igualmente elevada, de 0,9997. Esses resultados indicam que ambas as arquiteturas conseguiram discriminar adequadamente a classe minoritária mesmo em um cenário fortemente desbalanceado, mantendo simultaneamente alta precisão e alta sensibilidade. O *APPNP* apresentou desempenho intermediário, enquanto *GCN* e *GAT* tiveram resultados substancialmente inferiores, com menor capacidade de recuperação dos desinformadores e menor equilíbrio entre precisão e *recall*.

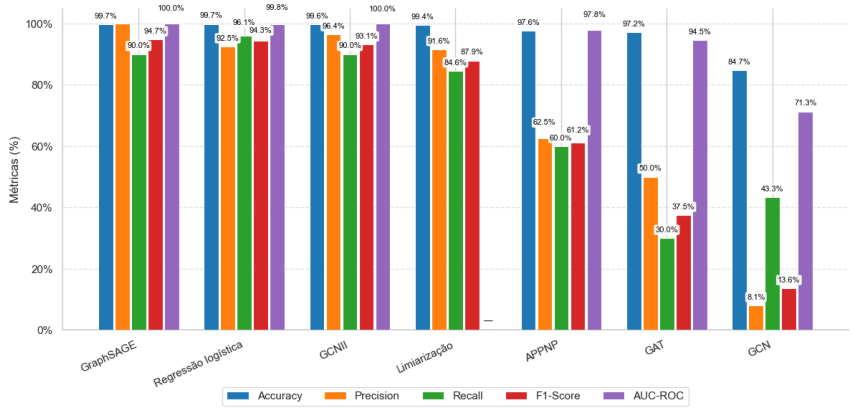

Figura 5. Comparação do Desempenho dos Modelos Avaliados.

A interpretação conjunta das métricas mostra diferenças relevantes entre as arquiteturas. Em um problema com apenas 2,14% de exemplos positivos, a *Accuracy* isoladamente tende a ser pouco informativa. Por isso, *Precision*, *Recall* e *F1-Score* são mais adequados para avaliar a utilidade prática dos classificadores. Nesse aspecto, o *GraphSAGE* se destacou por combinar alta precisão com *recall* elevado, resultando no maior *F1-Score* entre os modelos avaliados. O *GCNII* apresentou comportamento semelhante, embora ligeiramente inferior no equilíbrio entre recuperação e precisão. Já o *APPNP*, apesar de manter desempenho aceitável, mostrou queda mais acentuada nessas métricas. Os resultados de *GCN* e *GAT* indicam maior dificuldade para modelar adequadamente a estrutura da rede e separar usuários desinformadores dos não desinformadores no cenário considerado.

Para comparação direta com [Cabral et al. 2023], utilizou-se também uma divisão fixa de 70% para treino, 15% para validação e 15% para teste, apresentada na Tabela 4. Nessa configuração, o *GraphSAGE* apresentou o melhor desempenho geral entre os modelos avaliados neste trabalho, com *Accuracy* de 0,9972, *Precision* de 1,0000, *Recall* de 0,9000, *F1-Score* de 0,9474 e *AUC-ROC* de 0,9999. Esse resultado mostra que, na partição analisada, todas as predições positivas realizadas pelo modelo corresponderam a usuários efetivamente rotulados como desinformadores, sem comprometer de forma relevante a capacidade de recuperação da classe positiva. O *GCNII* apresentou desempenho próximo, com *Accuracy* de 0,9963, *Precision* de 0,9643, *Recall* de 0,9000, *F1-Score* de 0,9310 e *AUC-ROC* de 0,9998, confirmando-se como a segunda melhor arquitetura no experimento.

Na comparação com os referenciais de [Cabral et al. 2023], a regressão logística apresentou o maior *Recall* entre todos os modelos, com 0,9610, indicando maior capacidade de recuperar desinformadores na partição fixa. No entanto, o *GraphSAGE* superou esse resultado em *Accuracy*, *Precision*, *F1-Score* e *AUC-ROC*, evidenciando melhor equilíbrio global entre detecção correta e redução de falsos positivos. O método de limiarização também apresentou desempenho competitivo, mas inferior às melhores arquiteturas de GNN nas métricas de equilíbrio. Em contraste, *GAT*, *APPNP* e, principalmente, *GCN* apresentaram perdas expressivas em *Precision*, *Recall* e *F1-Score*. O *GCN*, por exemplo, embora tenha mantido *Accuracy* de 0,8465, obteve *Precision* de 0,0807 e *F1-Score* de 0,1361, o que confirma que sua acurácia elevada não se traduz em boa

identificação da classe minoritária. Em conjunto, os resultados mostram que *GraphSAGE* e *GCNII* foram as arquiteturas mais eficazes para a identificação de desinformadores, com vantagem para o *GraphSAGE* no compromisso entre precisão, sensibilidade e estabilidade global.

Tabela 4. Desempenho dos Modelos Avaliados.

Modelo	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Limiarização	0,9940	0,9160	0,8460	0,8790	–
Regressão logística	0,9970	0,9250	**0,9610**	0,9430	0,9980
GraphSAGE	**0,9972**	**1,0000**	0,9000	**0,9474**	**0,9999**
GCNII	0,9963	0,9643	0,9000	0,9310	0,9998
GAT	0,9721	0,5000	0,3000	0,3750	0,9452
APPNP	0,9764	0,6250	0,6000	0,6122	0,9777
GCN	0,8465	0,0807	0,4333	0,1361	0,7129

6. Conclusão

Este trabalho apresentou e avaliou uma metodologia para identificar desinformadores em grupos públicos do *WhatsApp* por meio de redes neurais em grafos. A abordagem foi formulada como um problema de classificação de nós em um grafo dirigido e ponderado, no qual os usuários foram representados por atributos comportamentais extraídos da própria plataforma, e as arestas foram reconstruídas a partir da coocorrência em grupos. Os resultados mostraram que a modelagem relacional da rede é informativa para a tarefa, pois arquiteturas que exploram explicitamente a propagação de informação entre vizinhos, em especial *GraphSAGE* e *GCNII*, apresentaram os melhores desempenhos no cenário avaliado. Em particular, o *GraphSAGE* apresentou os resultados mais consistentes no equilíbrio entre *precision*, *recall*, *F1-score* e *AUC-ROC*, enquanto o *GCNII* obteve desempenho próximo, confirmando a adequação de arquiteturas mais robustas à propagação em grafos densos e desbalanceados.

Os resultados também indicam que a incorporação da estrutura topológica da rede pode oferecer vantagens sobre abordagens que operam apenas sobre atributos individuais dos usuários. Mesmo em um cenário desbalanceado, com apenas 2,46% de usuários rotulados como desinformadores, os melhores modelos conseguiram manter alta capacidade discriminativa, o que mostra a viabilidade do uso de GNN para esse problema. Assim, além de comparar diferentes arquiteturas, este trabalho estabelece uma metodologia para o estudo da detecção de desinformadores em ambientes de mensageria instantânea, em que as relações entre usuários não são explicitamente observáveis e precisam ser inferidas a partir da dinâmica de participação em grupos. O código está disponível publicamente[6].

Como trabalhos futuros, recomenda-se ampliar a base de dados para diferentes períodos, contextos e eventos, a fim de avaliar a robustez do método em cenários menos polarizados. Também é relevante investigar estratégias de rotulação menos dependentes da força viral, incorporando atributos comportamentais, temporais e semânticos que permitam caracterizar com maior precisão diferentes perfis de disseminação.

7. Declaração sobre o uso de IA generativa

Não antecipamos preocupações sociais ou éticas imediatas decorrentes deste trabalho. Aelm disso, reconhecemos o uso de LLMs para auxiliar na revisão gramatical de partes do manuscrito.

[6]GitHub

Referências

Bondy, J. A., Murty, U. S. R., et al. (1976). *Graph theory with applications*, volume 290. Macmillan London.

Cabral, L., Martins, A. D. F., Monteiro, J. M., Machado, J. C., and Franco, W. (2023). Fakespreaderswhatsapp. br: Misinformation spreaders detection in brazilian portuguese whatsapp messages. In *ICEIS (1)*, pages 219–228.

Cabral, L., Monteiro, J. M., da Silva, J. W. F., Mattos, C. L. C., and Mourao, P. J. C. (2021). Fakewhastapp. br: Nlp and machine learning techniques for misinformation detection in brazilian portuguese whatsapp messages. In *ICEIS (1)*, pages 63–74.

Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. (2010). Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th annual computer security applications conference*, pages 21–30.

DeVerna, M. R., Aiyappa, R., Pacheco, D., Bryden, J., and Menczer, F. (2024). Identifying and characterizing superspreaders of low-credibility content on twitter. *Plos one*, 19(5):e0302201.

Easley, D., Kleinberg, J., et al. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*, volume 1. Cambridge university press Cambridge.

Ferreira, D. A., da Silva, A. P., and de Albuquerque Montenegro, C. (2021a). O impacto das fake news na vacinação e nos surtos de doenças erradicadas. *Revista Interdisciplinar em Saúde, Cajazeiras*, 8(8):2–16.

Ferreira, J. R. S., Lima, P. R. S., and Souza, E. D. d. (2021b). Desinformação, infodemia e caos social: impactos negativos das fake news no cenário da covid-19. *Em Questão*.

Fiocruz (2020). Pesquisa revela dados sobre fake news relacionadas à covid-19. Acesso em: 20 ago. 2025.

Khan, E. M., Ram, A., Rath, B., Vraga, E., and Srivastava, J. (2023). Behavioral forensics in social networks: Identifying misinformation, disinformation and refutation spreaders using machine learning. *arXiv preprint arXiv:2305.00957*.

Metzler, H. and Garcia, D. (2024). Social drivers and algorithmic mechanisms on digital media. *Perspectives on Psychological Science*, 19(5):735–748.

Nobre, G. P., Ferreira, C. H., and Almeida, J. M. (2022). A hierarchical network-oriented analysis of user participation in misinformation spread on whatsapp. *Information Processing & Management*, 59(1):102757.

Pereira, G., Bueno Bojczuk Camargo, I., and Parks, L. (2022). Whatsapp disruptions in brazil: A content analysis of user and news media responses, 2015–2018. *Global Media and Communication*, 18(1):113–148.

Rath, B., Salecha, A., and Srivastava, J. (2022). Fake news spreader detection using trust-based strategies in social networks with bot filtration. *Social network analysis and mining*, 12(1):66.

Raudeliuniene, J., Albats, E., and Kordab, M. (2021). Impact of information technologies and social networks on knowledge management processes in middle eastern audit and consulting companies. *Journal of Knowledge Management*, 25(4):871–898.

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Su, Q., Wan, M., Liu, X., Huang, C.-R., et al. (2020). Motivations, methods and metrics of misinformation detection: an nlp perspective. *Natural Language Processing Research*, 1(1-2):1–13.

Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *science*, 359(6380):1146–1151.

West, D. B. et al. (2001). *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.